

前端管理

bff@todo

我们的开源 欢迎star

<https://github.com/hug-sun/element3>

红宝书阅读计划

<https://www.bilibili.com/video/BV1SA411j7cB/>

到35岁如果还没做到管理是不是要考虑转行了？我参加工作比较晚，大学毕业后在部队待了五年，16年底左右才开始前端工作，今年33了。

我想转后端

我想转产品**

北京，薪资过勒30K，在考虑转

乾坤是做什么的

BFF @todo

大圣老师 我同学在考高级管理证，这个证有必要吗

微前端、组件化 怎么理解？怎么区别

缓存：load过一次，浏览器自然就缓存了

用微服务来解决啊，链接跳转还是方便的，但是，有些场景，就是不让你跳转，就是再一个应用里 统一的导航来做

管理

1. 赶鸭子上架，大部分同学担心自己的能力不行，担心自己的技术不是组内最好的
 1. 管理和前端架构师的能力体系，有很大的不同
 2. 一直做技术，转架构，转管理需要懂业务，懂运营，懂产品，懂撕逼，懂管理，会规划》。。。软技能
 3. python的作者入职微软了，汇报一个经理，他们组还有ts的作者，eclipse，设计模式黑皮书的作者等人，纯技术和技术管理
 4. 开课吧web组讲师招聘，招比我技术好的
 5. 尝试去申请资源，让这些程序员，很舒服的写代码。向上和向下

技术不能落伍的同事，要掌握横向的软技能 技术管理

职业生涯（核心竞争力是什么）

1. 纯技术，蛮辛苦，需要一些天赋 比较稳
2. 纯管理，比较危险，本身我们技术出身 行业绑定
3. 技术管理！！推荐的
4. 自由职业

脱离舒适区

9点45回来

选书

1. 求推荐 各种人推荐 《硅谷蓝图》
2. 你喜欢作者的别的书
3. 你喜欢的书里推荐的书
4. 关注几个知乎或者微博 关于推荐书的问题
5. 收藏几个豆瓣书单

不太建议屯书，随看随买

看一遍不想复习的，就直接扔多抓鱼卖了，一周左右就想复习的，一直放桌子上，一个月以后再整的，放书架上 我也不借书，我喜欢写写画画

买本书，到的时候，多抓鱼申请一个时间卖掉

规划

1. 技术规划
 1. 前端技术团队大概怎么规划
 2. 业务支持 是基础 同时能够做好组内的技术规划
 3. framework
 1. 组件
 2. node工具库，框架
 3. 脚手架
 4.
 4. 开发流程
 1. github gitlab ci/cd
 2. 代码规范，log 规范...

- 5. 数据mock文档
 - 6. 性能, 监控
 - 1. 埋点
 - 2. 日志
 - 3. 异常
 - 4. 性能统计
 - 7. nocode (搭建系统)
 - 8. 热更新, 自动化发布..... 一切能够让你团队自动化的内容
 - 9. 内部的技术分享流程
2. 人员规划
- 1. 我心中优先级更高
 - 2. 梯度建设, 有思考, 有努力, 有张有弛, 比较好的节奏
 - 1. 你滚蛋了, 谁来替你, 培养好替你人
 - 2. 我有很多时间去思考团队的下一步, 很多事情, 我可以分给老杨, 老夏,
 - 3. 你有核心竞争力, 并且梯度建设的好
 - 4. smart原则
 - 1. S目标明确 做一个vue3的ui框架, 带着学员做为目的
 - 2. m目标可量化: 3000star
 - 3. a 可达到
 - 4. m 和老板的计划相关
 - 5. t 达成的限制: 12月底
 - 3. 每个人员, 技能纬度的规划
 - 4. 导师制度的规划
 - 3. 可选项: 影响力 软技能
 - 1. 业务规划, 老板规划
 - 2. 标准规划, w3c, 贺老的tc39讨论
 - 3. 开源: element3, element3-admin, caixukun, 明年q1还回规划其他开源, React相关的

合格的管理者: 你再的时候, 效率, 产出逐步提升, 人员稳步发展, 你走了, 团队继续转
所有的新知识, 都会忘

Q3: vue代码贡献, 全员上王者

团队Q4的规划: 影响力, 帮助社区内更多的前端学习js or css or node

前端高级工程师 =》 带队的

讨论互动

微前端(训练营里都有)

介绍，训练营 这周就开通啦

1. 概念
2. qiankun实战 (阿里 dva, umi作者的团队)
3. 手写一个mini的微前端框架 -- caixukun

多个应用共存一个页面，最早用iframe，针对体验升级，针对spa微前端的架构

我们去饭馆吃饭，iframe就是来多少人，开多少包间，

只有一个包间，因为浏览器当前只有一个应用，页面跳转，来新客人，现在吃饭的人，收拾好桌子，暂时撤退，每次切应用都这么整 就是微前端

vue2, vue1, angular, react, 打包之后，都是一个html，内嵌引入css和js

1. html-loader 配置好上面4个应用的入口 解析对应的a应用的html，拿到他们依赖的css和js
2. 判断路由，决定执行哪个应用，load css和js，应用就启动
3. 路由切换，换一个应用b，干掉a加载进来的所有内容，包括css, js。。。
4. load新应用的css js

手写一个实现

多应用共存，新老版本应用共存

怎么去掉a应用加载的css效果，标记style标签，动态移除 shadow dom

怎么去掉a应用加载的js执行后的效果，js可能会修改全局变量，怎么恢复，用proxy代理window对象，记住修改了哪些，还可以用window快照，全量diff

1. 按需加载
2. 公共依赖加载
3. 预加载
4. 缓存。。。

前端未来

作业：

关于微前端和前端管理的思维导图

深夜话题

后端总是想领导前端怎么办

干他

怎样看，领导同时带后端和前端。在一个大组，领导也觉得前端简单，前端处于鄙视链的最下层。

这就是你如果作为前端leader需要做的事情，让他意识到不简单，做出成绩，最简单的方式，就是前端能够独立做点东西，争夺话语权

新技术，同事不想学，我又想用，怎么平衡比较好

自己私底下用，如果你是leader 听你的，如果你不是，自己的项目偷偷用

怎么撕产品，产品和领导，一直瞎搞，们开发都快疯了

撕逼是个技术活，如果一个产品不会前端，排期我就忽悠他

和产品更好的聊天，去学学产品，想和运营撕逼，学学运营

读读《毛选》

大龄程序员怎么保持竞争力

区别于年轻人，保持竞争力的方式 就是看书

当你想把产品的交互效果往用户体验更好的方向做的时候，后端就会跟你说甲方乙方的关系的想法，这个时候要怎么办？

听甲方爸爸的，你的体验的想法，私底下做就可以，我们有github，做了好东西，一定是有收获的

工作项目里的积累，可能还不如github上多积累的star

最近出去面试 感觉压工资的力度要比以往大得多啊 是不是等过完年再跳会好些？

年前跳，谈判空间大，损失年终奖

年后跳，你的竞争者变多，获得了年终奖，谈判空间变小

今年疫情，我的建议，工资压的多，就好好学习，年后再看，从现在，到过年，两个多月，足矣修炼一个很亮点的技能，打透

1. vue/react源码
2. 算法

web component

